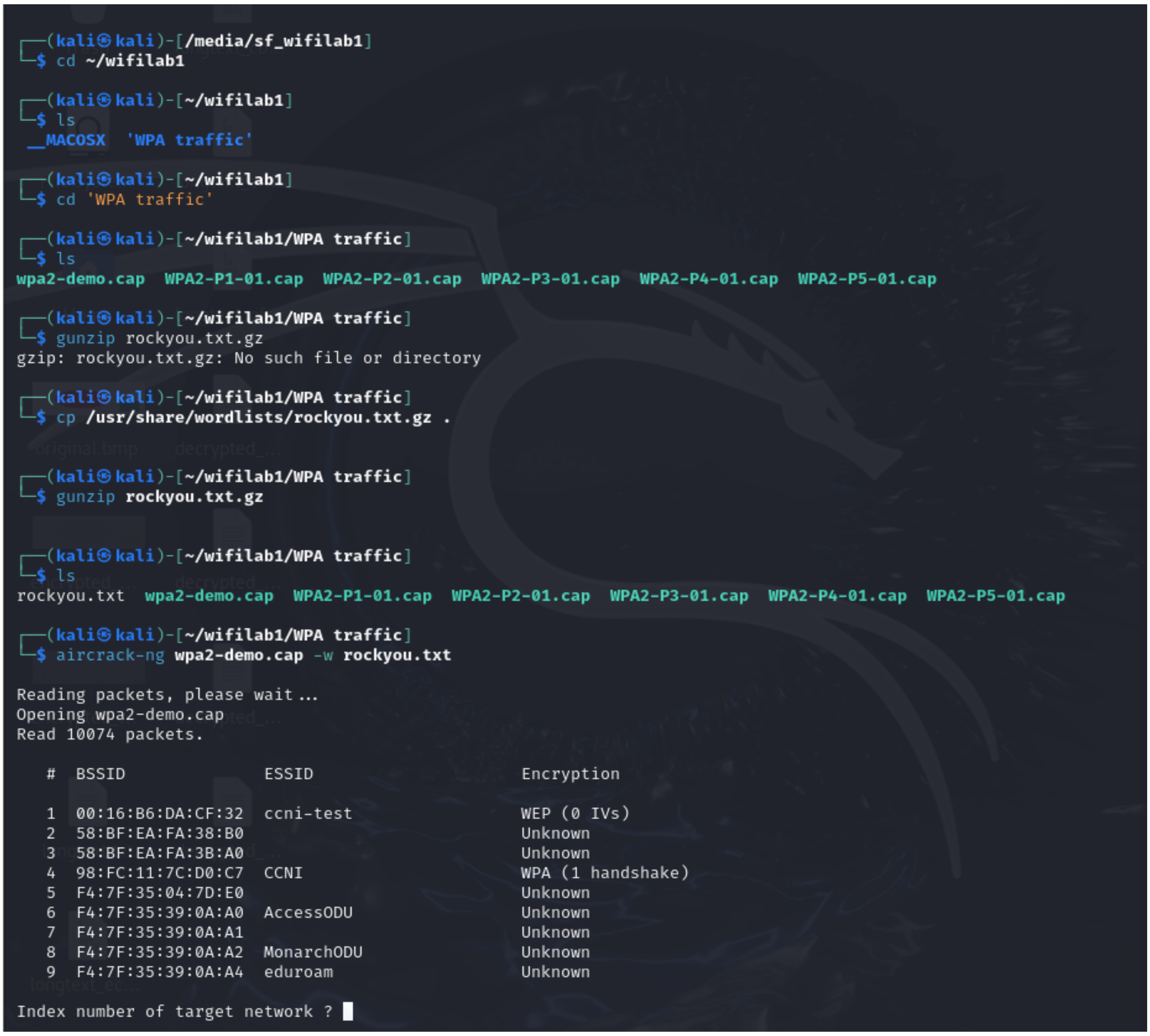
Lab 2 - WPA Security

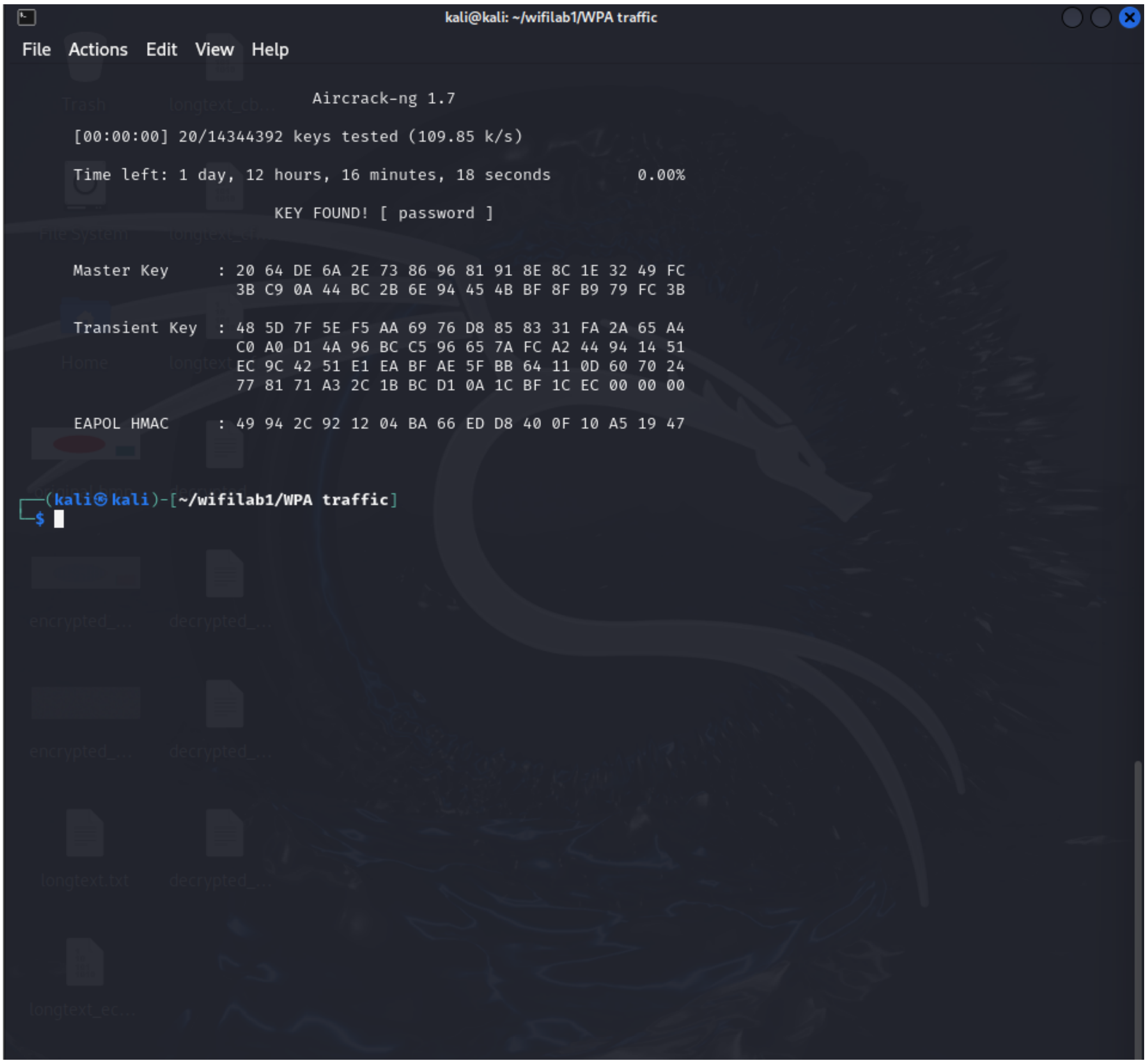
CYBR 8410: Distributed Systems Security

Task 1: In this task, I worked on decrypting a sample WPA2 WiFi capture file and analyzing the network traffic. First, I used a tool called aircrack-ng along with the popular rockyou.txt password list to crack the password from the file named wpa2-demo.cap. The password that was successfully found was password. Once I had the password, I used another tool called airdecap-ng to decrypt the traffic using that password and the Wi-Fi network name ESSID, which was CCNI. This gave me a new file called wpa2-demo-dec.cap, which contained the readable network activity



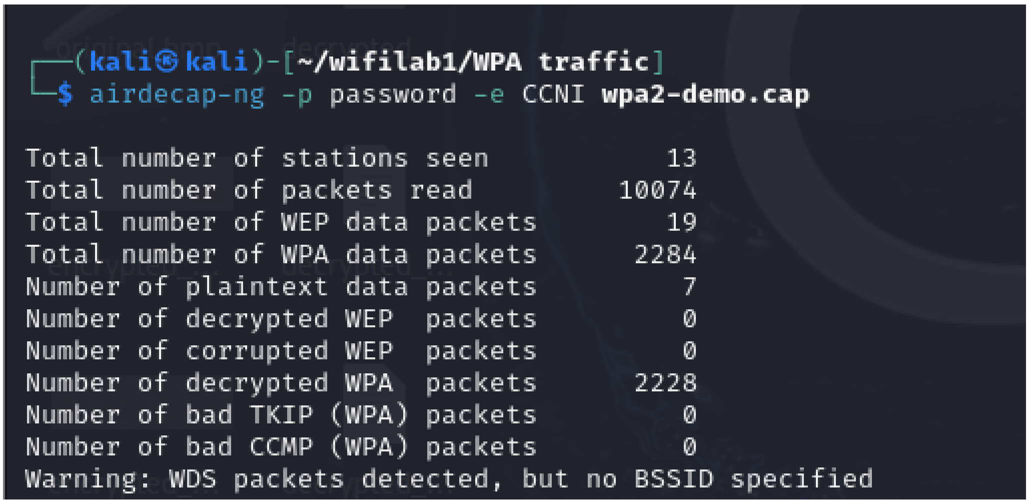
```
┌──(kali㉿kali)-[/media/sf_wifilab1]
└─$ cd ~/wifilab1

┌──(kali㉿kali)-[~/wifilab1]
└─$ ls
 __MACOSX  'WPA traffic'

┌──(kali㉿kali)-[~/wifilab1]
└─$ cd 'WPA traffic'

┌──(kali㉿kali)-[~/wifilab1/WPA traffic]
└─$ ls
wpa2-demo.cap  WPA2-P1-01.cap  WPA2-P2-01.cap  WPA2-P3-01.cap  WPA2-P4-01.cap  WPA2-P5-01.cap

┌──(kali㉿kali)-[~/wifilab1/WPA traffic]
└─$ gunzip rockyou.txt.gz
gzip: rockyou.txt.gz: No such file or directory

┌──(kali㉿kali)-[~/wifilab1/WPA traffic]
└─$ cp /usr/share/wordlists/rockyou.txt.gz .

┌──(kali㉿kali)-[~/wifilab1/WPA traffic]
└─$ gunzip rockyou.txt.gz


┌──(kali㉿kali)-[~/wifilab1/WPA traffic]
└─$ ls
rockyou.txt  wpa2-demo.cap  WPA2-P1-01.cap  WPA2-P2-01.cap  WPA2-P3-01.cap  WPA2-P4-01.cap  WPA2-P5-01.cap

┌──(kali㉿kali)-[~/wifilab1/WPA traffic]
└─$ aircrack-ng wpa2-demo.cap -w rockyou.txt

Reading packets, please wait...
Opening wpa2-demo.cap
Read 10074 packets.

   #  BSSID              ESSID                     Encryption

   1  00:16:B6:DA:CF:32  ccni-test                 WEP (0 IVs)
   2  58:BF:EA:FA:38:B0                            Unknown
   3  58:BF:EA:FA:3B:A0                            Unknown
   4  98:FC:11:7C:D0:C7  CCNI                      WPA (1 handshake)
   5  F4:7F:35:04:7D:E0                            Unknown
   6  F4:7F:35:39:0A:A0  AccessODU                 Unknown
   7  F4:7F:35:39:0A:A1                            Unknown
   8  F4:7F:35:39:0A:A2  MonarchODU                Unknown
   9  F4:7F:35:39:0A:A4  eduroam                   Unknown

Index number of target network ? █
```

Next I opened the decrypted file using Wireshark to examine the network traffic I observed various types of communication, including HTTP requests to sites google analytics.com. There were both GET and POST requests, but I did not find any usernames or passwords. I also found DNS queries to many domains showing which websites the user was trying to visit. Additionally I examined TCP traffic including some encrypted HTTPS connections and data exchanges with external IP addresses One of the internal IPs used was 192.168.1.127 and it was communicating with multiple public IP addresses.


kali@kali: ~/wifilab1/WPA traffic
File Actions Edit View Help

```
                               Aircrack-ng 1.7

      [00:00:00] 20/14344392 keys tested (109.85 k/s)

      Time left: 1 day, 12 hours, 16 minutes, 18 seconds         0.00%

                           KEY FOUND! [ password ]


      Master Key     : 20 64 DE 6A 2E 73 86 96 81 91 8E 8C 1E 32 49 FC
                       3B C9 0A 44 BC 2B 6E 94 45 4B BF 8F B9 79 FC 3B

      Transient Key  : 48 5D 7F 5E F5 AA 69 76 D8 85 83 31 FA 2A 65 A4
                       C0 A0 D1 4A 96 BC C5 96 65 7A FC A2 44 94 14 51
                       EC 9C 42 51 E1 EA BF AE 5F BB 64 11 0D 60 70 24
                       77 81 71 A3 2C 1B BC D1 0A 1C BF 1C EC 00 00 00

      EAPOL HMAC     : 49 94 2C 92 12 04 BA 66 ED D8 40 0F 10 A5 19 47


┌──(kali㉿kali)-[~/wifilab1/WPA traffic]
└─$ 
```

We successfully cracked the WPA password using aircrack-ng, and the key found was password. The capture file was processed correctly, and the handshake was valid

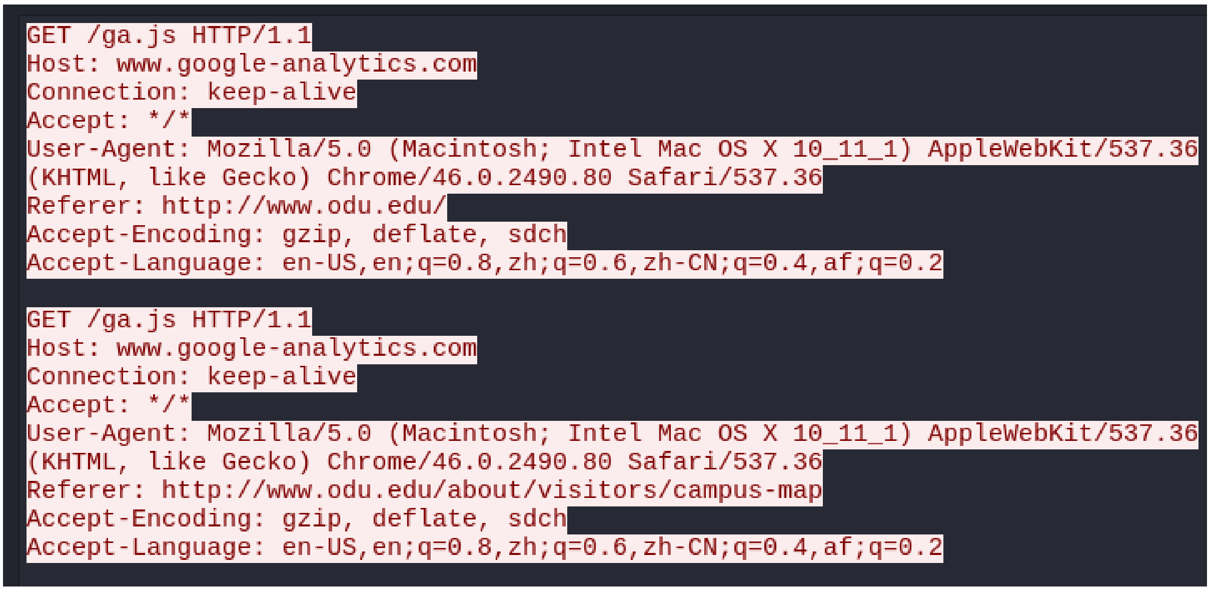


```
┌──(kali㉿kali)-[~/wifilab1/WPA traffic]
└─$ airdecap-ng -p password -e CCNI wpa2-demo.cap

Total number of stations seen            13
Total number of packets read          10074
Total number of WEP data packets         19
Total number of WPA data packets       2284
Number of plaintext data packets          7
Number of decrypted WEP  packets          0
Number of corrupted WEP  packets          0
Number of decrypted WPA  packets       2228
Number of bad TKIP (WPA) packets          0
Number of bad CCMP (WPA) packets          0
Warning: WDS packets detected, but no BSSID specified
```

This output from airdecap-ng confirms successful decryption of WPA packets using the provided password and ESSID CCNI from the wpa2-demo.cap file.

Overall, this task helped me learn how to crack a WiFi password decrypt the traffic and analyze different types of data such as web requests, DNS lookups and TCP connections. Even though I did not find any usernames or passwords in the traffic the analysis showed clear evidence of web browsing and app communication.

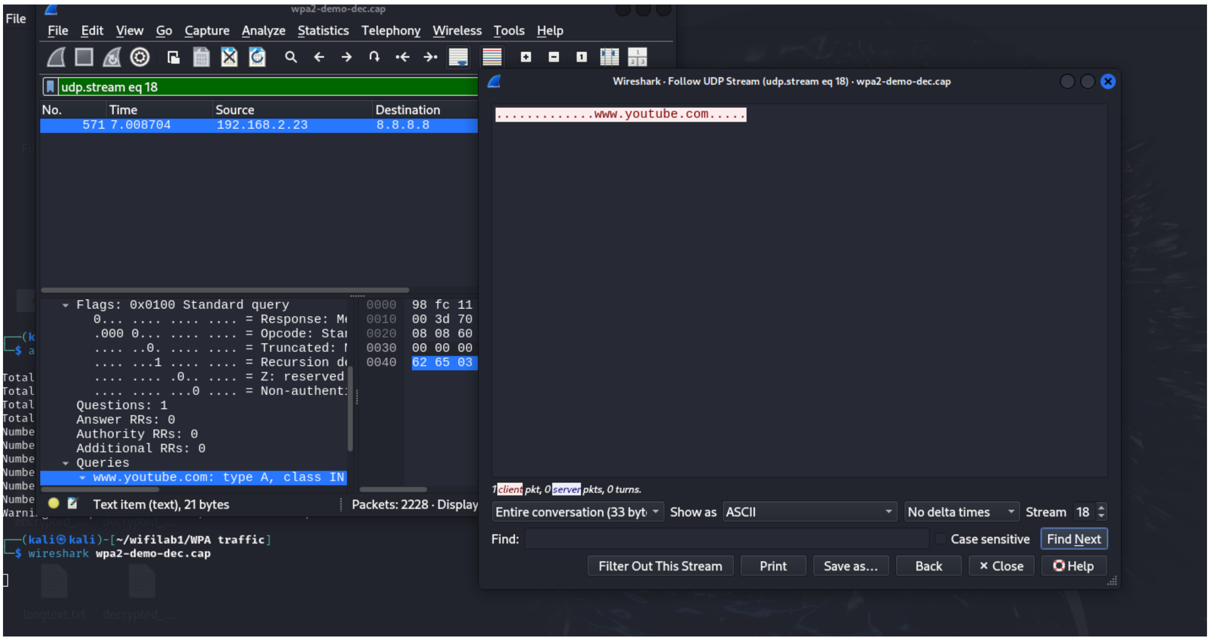
HTTP filter



```
GET /ga.js HTTP/1.1
Host: www.google-analytics.com
Connection: keep-alive
Accept: */*
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_1) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/46.0.2490.80 Safari/537.36
Referer: http://www.odu.edu/
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,zh;q=0.6,zh-CN;q=0.4,af;q=0.2

GET /ga.js HTTP/1.1
Host: www.google-analytics.com
Connection: keep-alive
Accept: */*
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_1) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/46.0.2490.80 Safari/537.36
Referer: http://www.odu.edu/about/visitors/campus-map
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,zh;q=0.6,zh-CN;q=0.4,af;q=0.2
```

I the HTTP I filtered the traffic using http to view unencrypted web activity. From the packets I observed several GET requests to google-analytics.com. The user was browsing a site that used this service for tracking. The browser useragent showed that the system was using macos and Chrome 46 and the Referer header pointed to pages from odu.edu indicating that the user visited that university website.

DNS filter



wpa2-demo-dec.cap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
udp.stream eq 18

No.   Time        Source          Destination
  571 7.008704    192.168.2.23    8.8.8.8

▾ Flags: 0x0100 Standard query                       0000  98 fc 11
    0... .... .... .... = Response: Me                0010  00 3d 70
    .000 0... .... .... = Opcode: Star                0020  08 08 60
    .... ..0. .... .... = Truncated: M                0030  00 00 00
    .... ...1 .... .... = Recursion de                0040  62 65 03
    .... .... .0.. .... = Z: reserved
    .... .... ...0 .... = Non-authent:
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
▾ Queries
  ▾ www.youtube.com: type A, class IN

Text item (text), 21 bytes          Packets: 2228 · Display
```

```
┌──(kali㉿kali)-[~/wifilab1/WPA traffic]
└─$ wireshark wpa2-demo-dec.cap
```

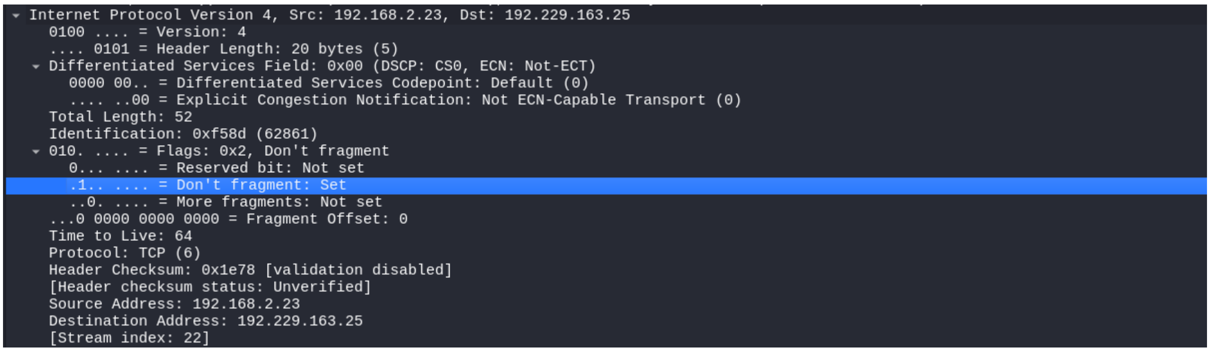
Wireshark · Follow UDP Stream (udp.stream eq 18) · wpa2-demo-dec.cap

```
.............www.youtube.com.....
```

1 client pkt, 0 server pkts, 0 turns.

Entire conversation (33 byt Show as ASCII · No delta times · Stream 18

This is DNS filter Using the dns filter, I found domain name lookups this shows what websites the user tried to visit. For example there was a dns query for www.youtube.com showing the user attempted to access YouTube Even though the actual content may be encrypted this query gives insight into browsing behavior.



```
▾ Internet Protocol Version 4, Src: 192.168.2.23, Dst: 192.229.163.25
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▾ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 52
    Identification: 0xf58d (62861)
  ▾ 010. .... = Flags: 0x2, Don't fragment
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: TCP (6)
    Header Checksum: 0x1e78 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.2.23
    Destination Address: 192.229.163.25
    [Stream index: 22]
```

I also reviewed packet details under the ip protocol. The internal IP address was 192.168.2.23, and I found external destinations like 192.229.163.25. The IP header included flags like Don't Fragment which tells routers not to break the packet into smaller pieces a normal part of IP traffic.

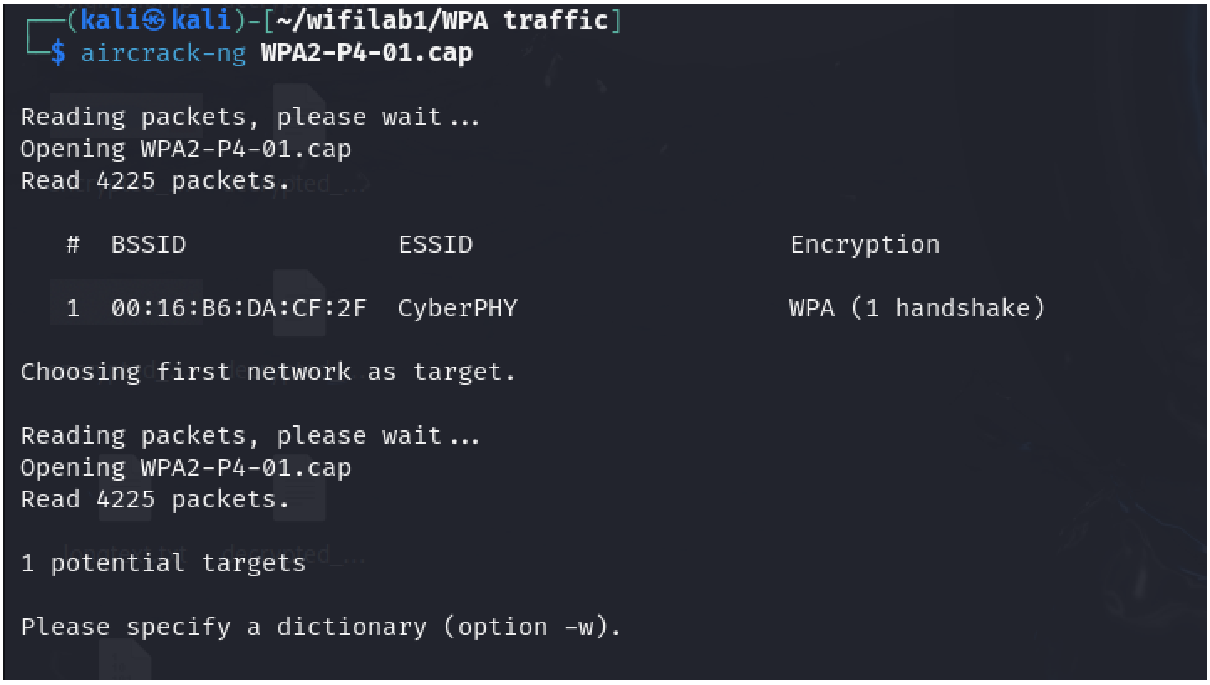
Summary Conclusion for Task A

From this decrypted traffic I was able to identify what websites were being visited what browser and operating system were used, and which domains were being looked up. However I did not find any login credentials or sensitive user data in the HTTP traffic. The information collected mostly reveals general web activity, like visiting educational sites or watching YouTube.

Task 2: For Task B, I was assigned the WPA2 capture file WPA2-P4-01.cap. First I used aircrack-ng and the rockyou.txt wordlist to run a dictionary attack on the capture file The correct Wi-Fi password was successfully found and it was linkinpark. After identifying the password I decrypted the capture file using airdecap-ng and the correct ESSID This produced a decrypted file named WPA2-P4-01-dec.cap which I then analyzed in Wireshark.

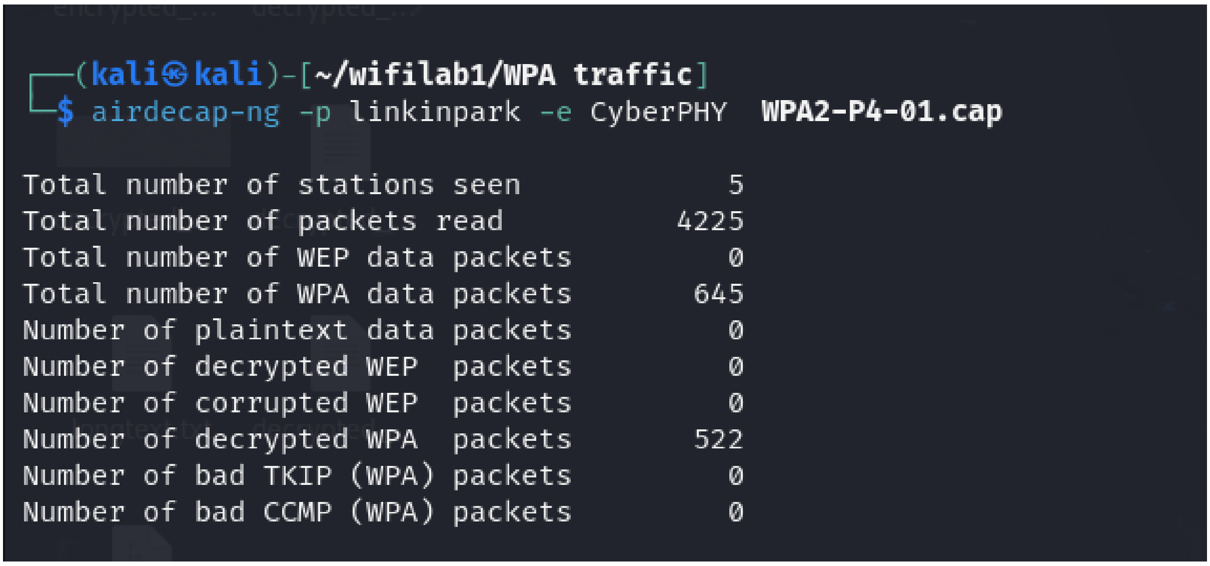
Inside the decrypted traffic I found a variety of user activity There were multiple HTTP GET and POST requests sent to different web servers. Some of these requests were to services like ajax.googleapis.com streamoc.music.tc.qq.com, and oth.eve.mdt.qq.com. The GET requests included media file downloads such as .m4a audio files while the POST requests appeared to be sending analytics or tracking data. I also found a HEAD request to www.taobao.com which indicates the client was checking a websites availability. Although these HTTP requests revealed browsing behavior and app activity, I did not find any usernames or passwords in the traffic.

I also examined DNS traffic using filters in Wireshark. The device was performing name resolution for several domains such as ogs.google.com eventlog.beacon.qq.com, and xiaomi.net. One of the DNS packets showed that www.taobao.com was resolved through a CNAME record to a cache server, demonstrating how websites use content delivery networks. In addition I analyzed TCP traffic and found HTTPS sessions over port 443. While the content of those sessions was encrypted, domain names like yahoo.com and gemini.yahoo.com were visible in the initial handshake exposing some user behavior.



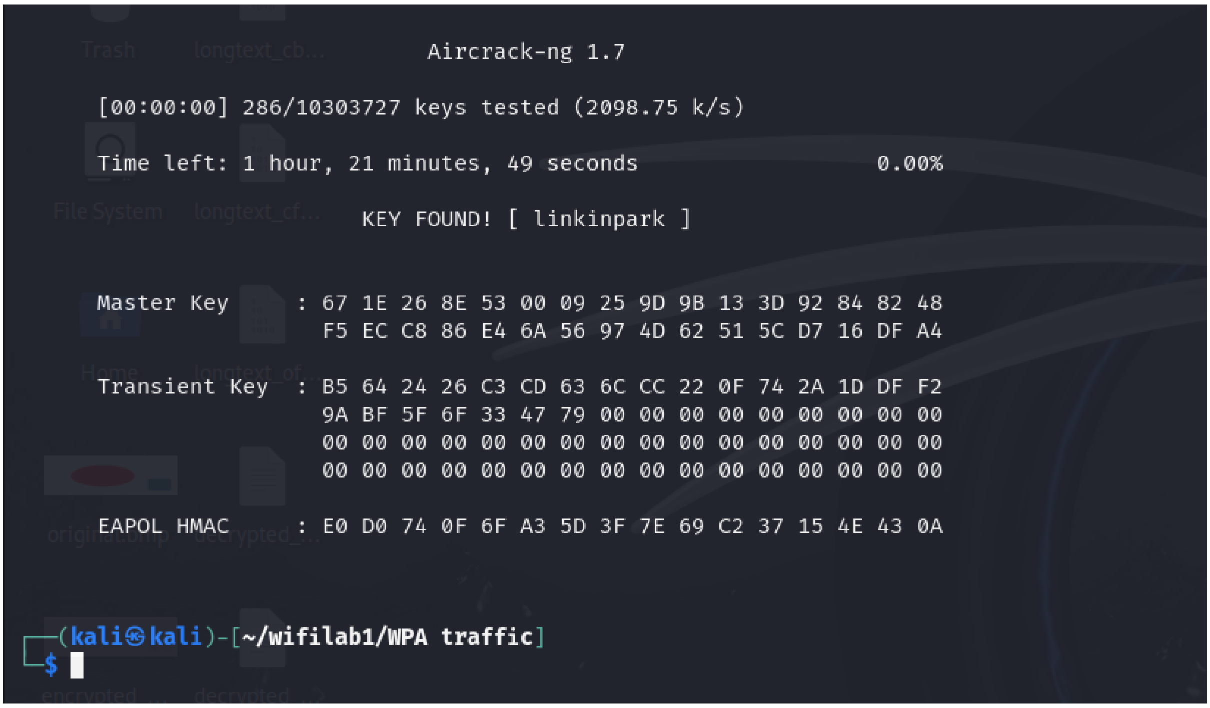
```
┌──(kali㉿kali)-[~/wifilab1/WPA traffic]
└─$ aircrack-ng WPA2-P4-01.cap

Reading packets, please wait...
Opening WPA2-P4-01.cap
Read 4225 packets.

   #  BSSID              ESSID                     Encryption

   1  00:16:B6:DA:CF:2F  CyberPHY                  WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening WPA2-P4-01.cap
Read 4225 packets.

1 potential targets

Please specify a dictionary (option -w).
```

The aircrack-ng command was run on WPA2-P4-01.cap, which contained 4225 packets and identified a Wi-Fi network named CyberPHY with one WPA handshake.



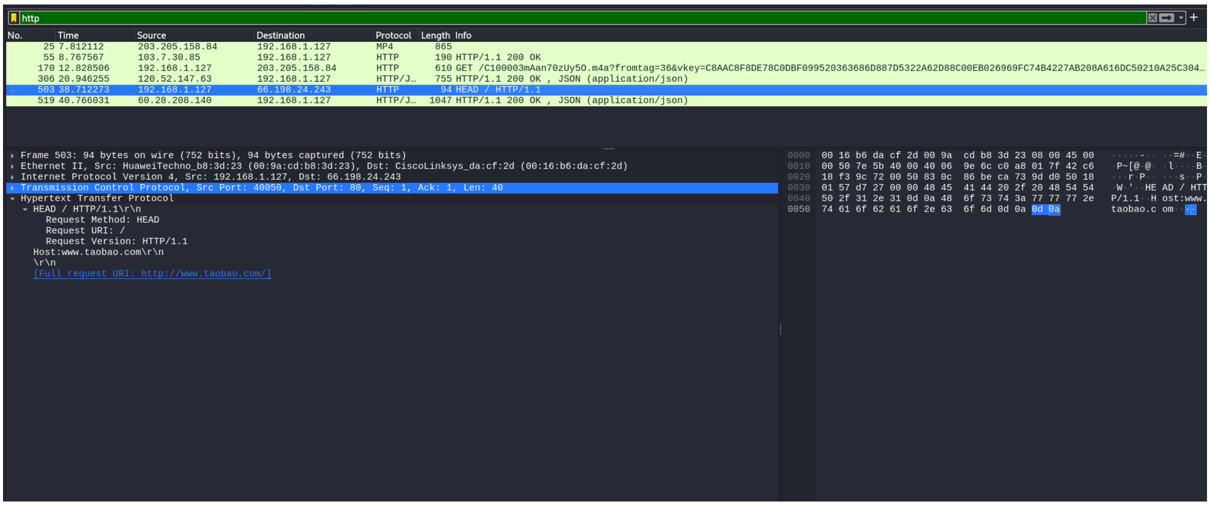
```
┌──(kali㉿kali)-[~/wifilab1/WPA traffic]
└─$ airdecap-ng -p linkinpark -e CyberPHY  WPA2-P4-01.cap

Total number of stations seen            5
Total number of packets read          4225
Total number of WEP data packets         0
Total number of WPA data packets       645
Number of plaintext data packets         0
Number of decrypted WEP  packets         0
Number of corrupted WEP  packets         0
Number of decrypted WPA  packets       522
Number of bad TKIP (WPA) packets         0
Number of bad CCMP (WPA) packets         0
```

the successful decryption of WPA2 traffic using the password linkinpark and ESSID CyberPHY with the airdecap-ng tool.



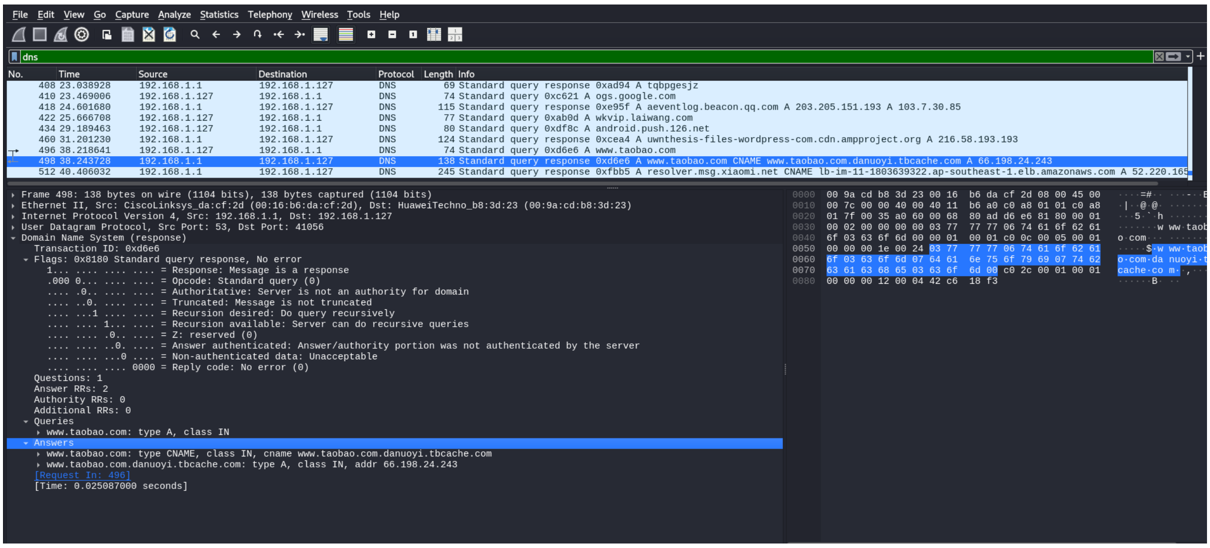
```
                               Aircrack-ng 1.7

      [00:00:00] 286/10303727 keys tested (2098.75 k/s)

      Time left: 1 hour, 21 minutes, 49 seconds                  0.00%

                          KEY FOUND! [ linkinpark ]


      Master Key     : 67 1E 26 8E 53 00 09 25 9D 9B 13 3D 92 84 82 48
                       F5 EC C8 86 E4 6A 56 97 4D 62 51 5C D7 16 DF A4

      Transient Key  : B5 64 24 26 C3 CD 63 6C CC 22 0F 74 2A 1D DF F2
                       9A BF 5F 6F 33 47 79 00 00 00 00 00 00 00 00 00
                       00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                       00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

      EAPOL HMAC     : E0 D0 74 0F 6F A3 5D 3F 7E 69 C2 37 15 4E 43 0A


┌──(kali㉿kali)-[~/wifilab1/WPA traffic]
└─$ 
```

confirms that the WPA handshake was successfully cracked using Aircrack-ng, revealing the Wi-Fi password as **linkinpark**.



```
http
No.  Time       Source          Destination     Protocol Length Info
  25 7.812112   203.205.158.84  192.168.1.127   MP4        865
  55 8.767567   103.7.30.85     192.168.1.127   HTTP       190 HTTP/1.1 200 OK
 170 12.828506  192.168.1.127   203.205.158.84  HTTP       610 GET /C100003mAan70zUy5O.m4a?fromtag=36&vkey=C8AAC8F8DE78C0DBF099520363686D887D5322A62D88C00EB026969FC74B4227AB208A616DC50210A25C304…
 306 20.946255  120.52.147.63   192.168.1.127   HTTP/J…    755 HTTP/1.1 200 OK , JSON (application/json)
 503 38.712273  192.168.1.127   66.198.24.243   HTTP        94 HEAD / HTTP/1.1
 519 40.766031  60.28.208.140   192.168.1.127   HTTP/J…   1047 HTTP/1.1 200 OK , JSON (application/json)

› Frame 503: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
› Ethernet II, Src: HuaweiTechno_b8:3d:23 (00:9a:cd:b8:3d:23), Dst: CiscoLinksys_da:cf:2d (00:16:b6:da:cf:2d)
› Internet Protocol Version 4, Src: 192.168.1.127, Dst: 66.198.24.243
› Transmission Control Protocol, Src Port: 40050, Dst Port: 80, Seq: 1, Ack: 1, Len: 40
⌄ Hypertext Transfer Protocol
  ⌄ HEAD / HTTP/1.1\r\n
      Request Method: HEAD
      Request URI: /
      Request Version: HTTP/1.1
    Host:www.taobao.com\r\n
    \r\n
    [Full request URI: http://www.taobao.com/]

0000  00 16 b6 da cf 2d 00 9a  cd b8 3d 23 08 00 45 00   ······-·· ··=#··E·
0010  00 50 7e 5b 40 00 40 06  9e 6c c0 a8 01 7f 42 c6   ·P~[@·@·  ·l····B·
0020  18 f3 9c 72 00 50 83 0c  86 be ca 73 9d d0 50 18   ···r·P··  ···s··P·
0030  01 57 d7 27 00 00 48 45  41 44 20 2f 20 48 54 54   ·W·'··HE AD / HTT
0040  50 2f 31 2e 31 0d 0a 48  6f 73 74 3a 77 77 77 2e   P/1.1··H ost:www.
0050  74 61 6f 62 61 6f 2e 63  6f 6d 0d 0a 0d 0a         taobao.c om····
```

This HTTP filter in this I have seen HTTP HEAD request to www.taobao.com made from IP 192.168.1.127 to destination 66.198.24.243

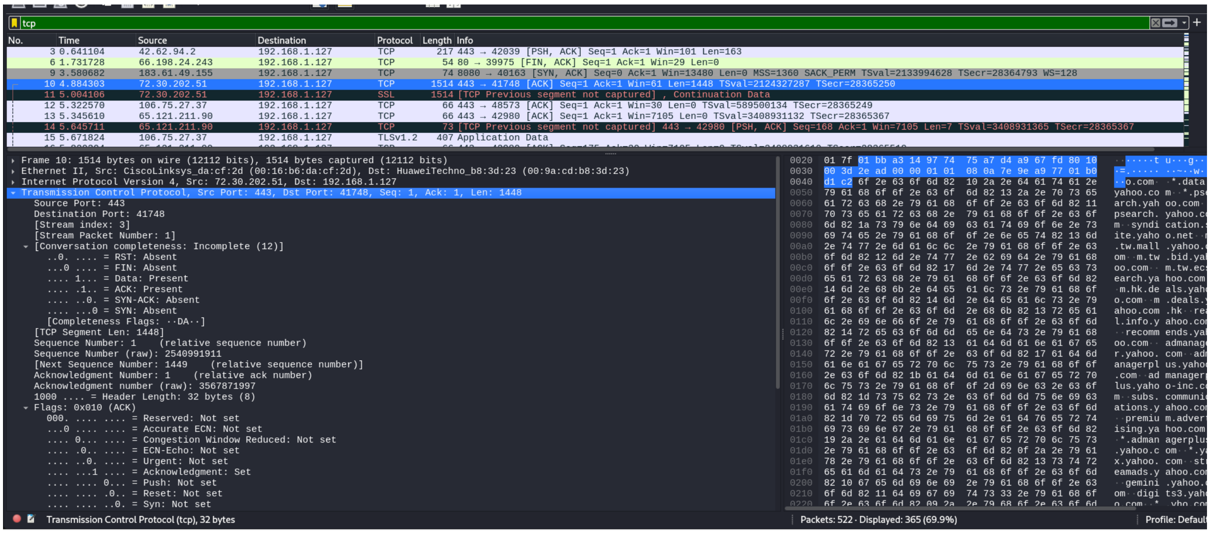


```
dns

No.   Time         Source          Destination     Protocol Length Info
  408 23.038928    192.168.1.1     192.168.1.127   DNS        69 Standard query response 0xad94 A tqbpgesjz
  410 23.469006    192.168.1.127   192.168.1.1     DNS        74 Standard query 0xc621 A ogs.google.com
  418 24.601680    192.168.1.1     192.168.1.127   DNS       115 Standard query response 0xe95f A aeventlog.beacon.qq.com A 203.205.151.193 A 103.7.30.85
  422 25.666708    192.168.1.127   192.168.1.1     DNS        77 Standard query 0xab0d A wkvip.laiwang.com
  434 29.189463    192.168.1.127   192.168.1.1     DNS        80 Standard query 0xdf8c A android.push.126.net
  460 31.201230    192.168.1.1     192.168.1.127   DNS       124 Standard query response 0xcea4 A uwnthesis-files-wordpress-com.cdn.ampproject.org A 216.58.193.193
  496 38.218641    192.168.1.127   192.168.1.1     DNS        74 Standard query 0xd6e6 A www.taobao.com
  498 38.243728    192.168.1.1     192.168.1.127   DNS       138 Standard query response 0xd6e6 A www.taobao.com CNAME www.taobao.com.danuoyi.tbcache.com A 66.198.24.243
  512 40.406032    192.168.1.1     192.168.1.127   DNS       245 Standard query response 0xfbb5 A resolver.msg.xiaomi.net CNAME lb-im-11-1803639322.ap-southeast-1.elb.amazonaws.com A 52.220.165
```

```
› Frame 498: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits)
› Ethernet II, Src: CiscoLinksys_da:cf:2d (00:16:b6:da:cf:2d), Dst: HuaweiTechno_b8:3d:23 (00:9a:cd:b8:3d:23)
› Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.127
› User Datagram Protocol, Src Port: 53, Dst Port: 41056
⌄ Domain Name System (response)
    Transaction ID: 0xd6e6
  ⌄ Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
  ⌄ Queries
    › www.taobao.com: type A, class IN
  ⌄ Answers
    › www.taobao.com: type CNAME, class IN, cname www.taobao.com.danuoyi.tbcache.com
    › www.taobao.com.danuoyi.tbcache.com: type A, class IN, addr 66.198.24.243
    [Request In: 496]
    [Time: 0.025087000 seconds]
```

```
0000  00 9a cd b8 3d 23 00 16  b6 da cf 2d 08 00 45 00   ····=#·· ···-··E
0010  00 7c 00 00 40 00 40 11  b6 a0 c0 a8 01 01 c0 a8   ·|··@·@· ········
0020  01 7f 00 35 a0 60 00 68  80 ad d6 e6 81 80 00 01   ···5·`·h ········
0030  00 02 00 00 00 00 03 77  77 77 06 74 61 6f 62 61   ·······w ww·taob
0040  6f 03 63 6f 6d 00 00 01  00 01 c0 0c 00 05 00 01   o·com··· ········
0050  00 00 00 1e 00 24 03 77  77 77 06 74 61 6f 62 61   ·····$·w ww·taob
0060  6f 03 63 6f 6d 07 64 61  6e 75 6f 79 69 07 74 62   o·com·da nuoyi·t
0070  63 61 63 68 65 03 63 6f  6d 00 c0 2c 00 01 00 01   cache·co m··,····
0080  00 00 00 12 00 04 42 c6  18 f3                     ······B· ··
```

This DNS response in Wireshark shows that the domain www.taobao.com resolves to a CNAME www.taobao.com.danuoyi.tbcache.com, which then resolves to the IP address 66.198.24.243. This indicates the true backend server serving taobao.com s content.



```
tcp

No.   Time        Source           Destination     Protocol Length Info
    3 0.641104    42.62.94.2       192.168.1.127   TCP       217 443 → 42039 [PSH, ACK] Seq=1 Ack=1 Win=101 Len=163
    6 1.731728    66.198.24.243    192.168.1.127   TCP        54 80 → 39975 [FIN, ACK] Seq=1 Ack=1 Win=29 Len=0
    9 3.580682    183.61.49.155    192.168.1.127   TCP        74 8080 → 40163 [SYN, ACK] Seq=0 Ack=1 Win=13480 Len=0 MSS=1360 SACK_PERM TSval=2133994628 TSecr=28364793 WS=128
   10 4.884303    72.30.202.51     192.168.1.127   TCP      1514 443 → 41748 [ACK] Seq=1 Ack=1 Win=61 Len=1448 TSval=2124327287 TSecr=28365250
   11 5.004106    72.30.202.51     192.168.1.127   SSL      1514 [TCP Previous segment not captured] , Continuation Data
   12 5.322570    106.75.27.37     192.168.1.127   TCP        66 443 → 48573 [ACK] Seq=1 Ack=1 Win=30 Len=0 TSval=589500134 TSecr=28365249
   13 5.345610    65.121.211.90    192.168.1.127   TCP        66 443 → 42980 [ACK] Seq=1 Ack=1 Win=7105 Len=0 TSval=3408931132 TSecr=28365367
   14 5.645711    65.121.211.90    192.168.1.127   TCP        73 [TCP Previous segment not captured] 443 → 42980 [PSH, ACK] Seq=168 Ack=1 Win=7105 Len=7 TSval=3408931365 TSecr=28365367
   15 5.671824    106.75.27.37     192.168.1.127   TLSv1.2   407 Application Data
```

```
› Frame 10: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
› Ethernet II, Src: CiscoLinksys_da:cf:2d (00:16:b6:da:cf:2d), Dst: HuaweiTechno_b8:3d:23 (00:9a:cd:b8:3d:23)
› Internet Protocol Version 4, Src: 72.30.202.51, Dst: 192.168.1.127
⌄ Transmission Control Protocol, Src Port: 443, Dst Port: 41748, Seq: 1, Ack: 1, Len: 1448
    Source Port: 443
    Destination Port: 41748
    [Stream index: 3]
    [Stream Packet Number: 1]
  ⌄ [Conversation completeness: Incomplete (12)]
      ..0. .... = RST: Absent
      ...0 .... = FIN: Absent
      .... 1... = Data: Present
      .... .1.. = ACK: Present
      .... ..0. = SYN-ACK: Absent
      .... ...0 = SYN: Absent
      [Completeness Flags: ··DA··]
    [TCP Segment Len: 1448]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 2540991911
    [Next Sequence Number: 1449    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 3567871997
    1000 .... = Header Length: 32 bytes (8)
  ⌄ Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Accurate ECN: Not set
      .... 0... .... = Congestion Window Reduced: Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
```

```
0020  01 7f 01 bb a3 14 97 74  75 a7 d4 a9 67 fd 80 10   ·······t u···g··
0030  00 3d 2e ad 00 00 01 01  08 0a 7e 9e a9 77 01 b0   ·=.····· ··~··w·
0040  d1 c2 6f 2e 63 6f 6d 82  10 2a 2e 64 61 74 61 2e   ··o.com· ·*.data
0050  79 61 68 6f 6f 2e 63 6f  6d 82 13 2a 2e 70 73 65   yahoo.co m··*.pse
0060  61 72 63 68 2e 79 61 68  6f 6f 2e 63 6f 6d 82 11   arch.yah oo.com··
0070  70 73 65 61 72 63 68 2e  79 61 68 6f 6f 2e 63 6f   psearch. yahoo.co
0080  6d 82 1a 73 79 6e 64 69  63 61 74 69 6f 6e 2e 73   m··syndi cation.s
0090  69 74 65 2e 79 61 68 6f  6f 2e 6e 65 74 82 13 6d   ite.yaho o.net··m
00a0  2e 74 77 2e 6d 61 6c 6c  2e 79 61 68 6f 6f 2e 63   .tw.mall .yahoo.c
00b0  6f 6d 82 12 6d 2e 74 77  2e 62 69 64 2e 79 61 68   om··m.tw .bid.yah
00c0  6f 6f 2e 63 6f 6d 82 17  6d 2e 74 77 2e 65 63 73   oo.com·· m.tw.ecs
00d0  65 61 72 63 68 2e 79 61  68 6f 6f 2e 63 6f 6d 82   earch.ya hoo.com·
00e0  14 6d 2e 68 6b 2e 64 65  61 6c 73 2e 79 61 68 6f   ·m.hk.de als.yaho
00f0  6f 2e 63 6f 6d 82 14 6d  2e 64 65 61 6c 73 2e 79   o.com··m .deals.y
0100  61 68 6f 6f 2e 63 6f 6d  2e 68 6b 82 13 72 65 61   ahoo.com .hk··rea
0110  6c 2e 69 6e 66 6f 2e 79  61 68 6f 6f 2e 63 6f 6d   l.info.y ahoo.com
0120  82 14 72 65 63 6f 6d 6d  65 6e 64 73 2e 79 61 68   ··recomm ends.yah
0130  6f 6f 2e 63 6f 6d 82 13  61 64 6d 61 6e 61 67 65   oo.com·· admanage
0140  72 2e 79 61 68 6f 6f 2e  63 6f 6d 82 17 61 64 6d   r.yahoo. com··adm
0150  61 6e 61 67 65 72 70 6c  75 73 2e 79 61 68 6f 6f   anagerpl us.yahoo
0160  2e 63 6f 6d 82 1b 61 64  6d 61 6e 61 67 65 72 70   .com··ad managerp
0170  6c 75 73 2e 79 61 68 6f  6f 2d 69 6e 63 2e 63 6f   lus.yaho o-inc.co
0180  6d 82 1d 73 75 62 73 2e  63 6f 6d 6d 75 6e 69 63   m··subs. communic
0190  61 74 69 6f 6e 73 2e 79  61 68 6f 6f 2e 63 6f 6d   ations.y ahoo.com
01a0  82 1d 70 72 65 6d 69 75  6d 2e 61 64 76 65 72 74   ··premiu m.advert
01b0  69 73 69 6e 67 2e 79 61  68 6f 6f 2e 63 6f 6d 82   ising.ya hoo.com·
01c0  19 2a 2e 61 64 6d 61 6e  61 67 65 72 70 6c 75 73   ·*.adman agerplus
01d0  2e 79 61 68 6f 6f 2e 63  6f 6d 82 0f 2a 2e 79 61   .yahoo.c om··*.ya
01e0  78 2e 79 61 68 6f 6f 2e  63 6f 6d 82 13 73 74 72   x.yahoo. com··str
01f0  65 61 6d 61 64 73 2e 79  61 68 6f 6f 2e 63 6f 6d   eamads.y ahoo.com
0200  82 10 67 65 6d 69 6e 69  2e 79 61 68 6f 6f 2e 63   ··gemini .yahoo.c
0210  6f 6d 82 11 64 69 67 69  74 73 33 2e 79 61 68 6f   om··digi ts3.yaho
```

Transmission Control Protocol (tcp), 32 bytes   Packets: 522 · Displayed: 365 (69.9%)   Profile: Default

This TCP packet capture shows a communication from IP 72.30.202.51 to 192.168.1.127 over port 443, indicating encrypted HTTPS traffic.

What We Did:

In this lab, we captured and decrypted WPA2 WiFi traffic using a .cap file and dictionary attack tools like Aircrack-ng.
We identified the network ESSID CyberPHY, CCNI and cracked the WiFi password successfully (linkinpark, password).
Using Wireshark, we filtered and analyzed HTTP, DNS, IP, and TCP traffic, looking for patterns, websites visited, and potential plaintext credentials.
We decrypted the captured packets using airdecap-ng and observed decoded traffic to identify important network behavior.
Finally, we followed streams and visualized packet spikes to detect traffic anomalies and confirmed no visible usernames/passwords were leaked.

What We Learned:

We learned how WPA2 handshakes can be captured and cracked using a dictionary attack with proper tools.
We practiced using Wireshark filters to analyze traffic types and dig into packet-level details.
We understood how to detect common traffic patterns, such as DNS queries, HTTP requests, and TCP session flows.
We learned to identify ESSID/BSSID from packet captures and how to decrypt them.
Most importantly, we gained hands-on experience in wireless traffic analysis and how poor encryption can expose sensitive data.